KEY_Practice08_Logic

December 10, 2021

1 Practice with logic

We will practice: - using conditional operators to create booleans - identifying conditional operators that are opposite - using keywords like not, is, and, or

Pretend we are having a Girls Who Code party! We want to order cupcakes and brownies for the party so that we can enjoy some sweet treats together.

At the start of our planning, we have 25 guests attending. We have already baked 10 cupcakes and 10 brownies.

[1]:
```
cupcakes = 10
brownies = 10

guests = 25
```

Everyone coming to the party wants at least one cupcake and at least one brownie. Therefore, to have enough cupcakes and brownies, we want the number of each to be *at least* the number of guests.

[7]:
```
# create a boolean that tells us if we have enough cupcakes; save in␣
↪enough_cupcakes
enough_cupcakes = cupcakes >= guests
# print enough_cupcakes
print(enough_cupcakes)

# create a boolean that tells us if we have enough brownies; save in␣
↪enough_brownies
enough_brownies = brownies >= guests
# print enough_brownies
print(enough_brownies)
```

```
False
False
```

We have enough food if we have enough cupcakes **and** we have enough brownies. Do we have enough food for the party?

[12]:
```
# create a boolean called enough_food that says if we are ready for the party
enough_food = enough_cupcakes and enough_brownies
```

```
# print enough_food
print(enough_food)
```

```
False
```

The night before the party, we get busy baking, and we make 20 more cupcakes. Are we ready now?

[13]:
```
# add 20 to cupcakes
cupcakes = cupcakes + 20

# re-calculate enough_cupcakes
enough_cupcakes = cupcakes >= guests

# re-calculate enough_food
enough_food = enough_cupcakes and enough_brownies

# print enough_food
print(enough_food)
```

```
False
```

We also want to order party hats for the party. Hats can be expensive, so we don't want to order more than we need, but we also want everyone to have one.

[11]:
```
# create a variable called hats and set it equal to 25
hats = 25

# create a boolean that determines if hats equals guests, and store it in␣
↪enough_hats
enough_hats = hats == guests

# print enough_hats
print(enough_hats)
```

```
True
```

We are ready for the party when we have enough food **and** when we have enough hats. Are we ready for the party?

[14]:
```
# create a variable called ready that says if we have enough food and enough␣
↪hats
ready = enough_food and enough_hats

# print ready
print(ready)
```

```
False
```

[18]:
```
# add 30 to brownies
brownies = brownies + 30

# re-calculate enough_brownies, enough_food and ready
enough_brownies = brownies >= guests
enough_food = enough_brownies and enough_cupcakes
ready = enough_food and enough_hats

# print ready
print(ready)
```

```
True
```